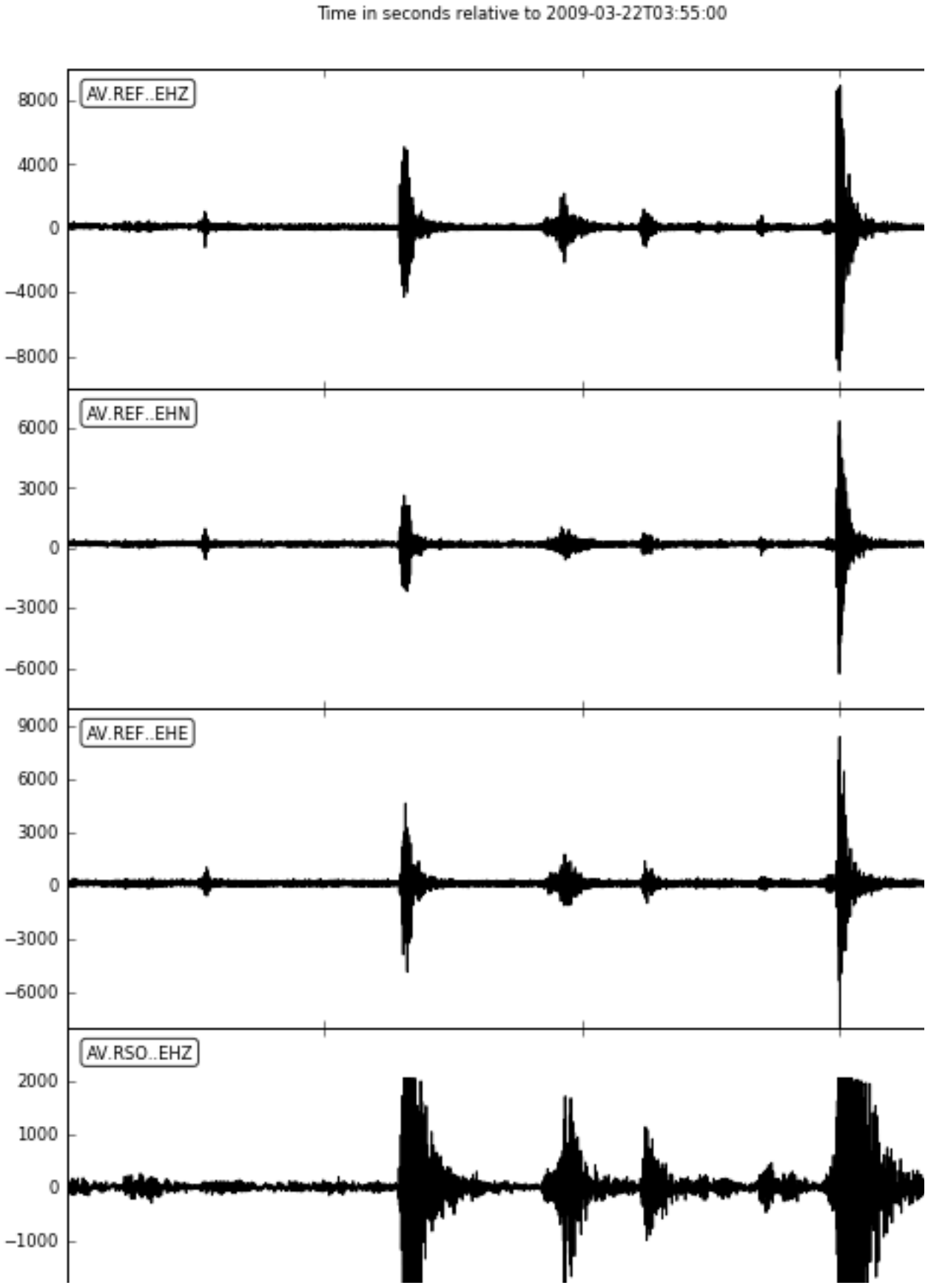
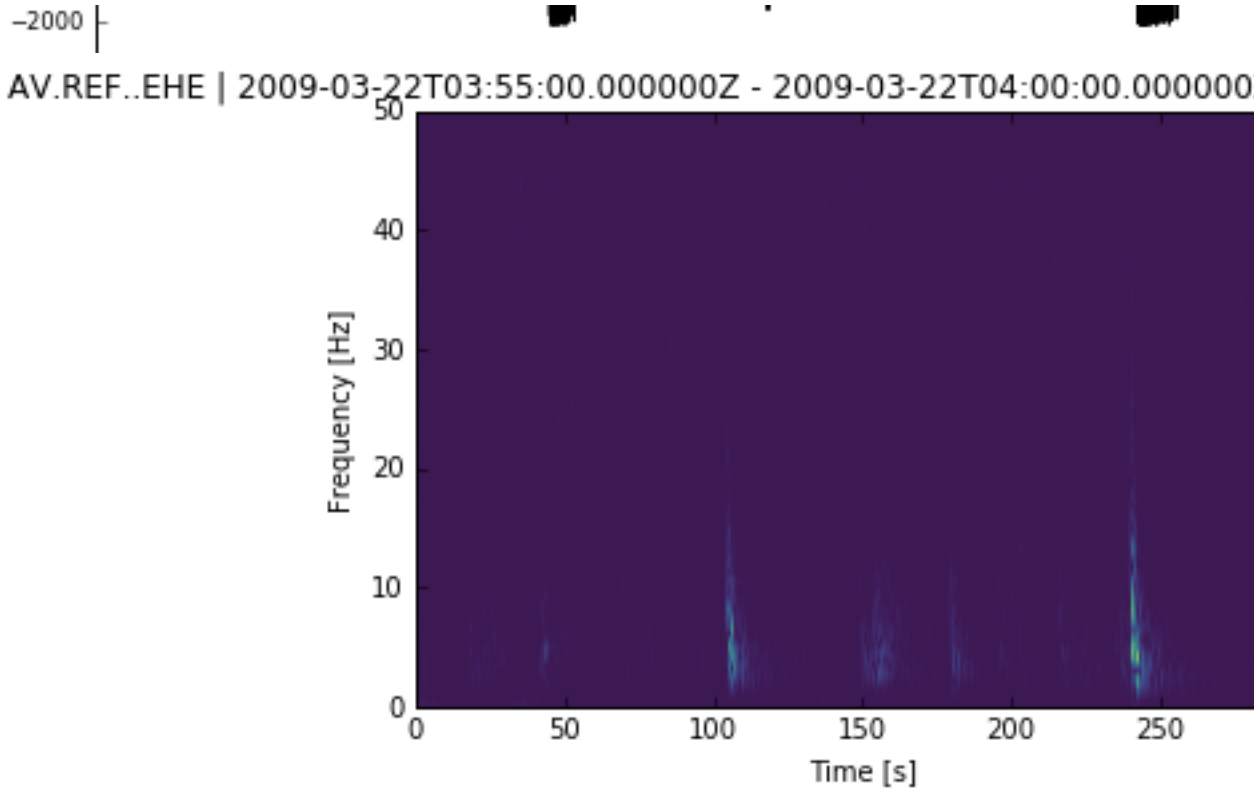
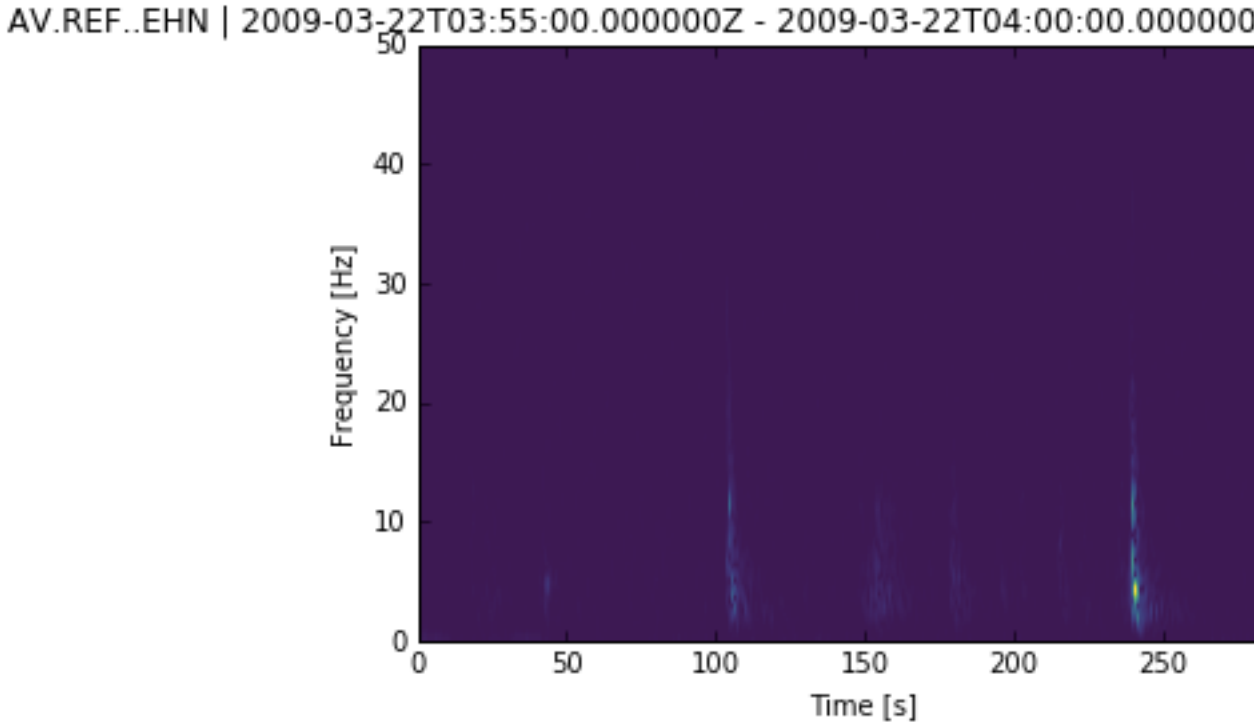
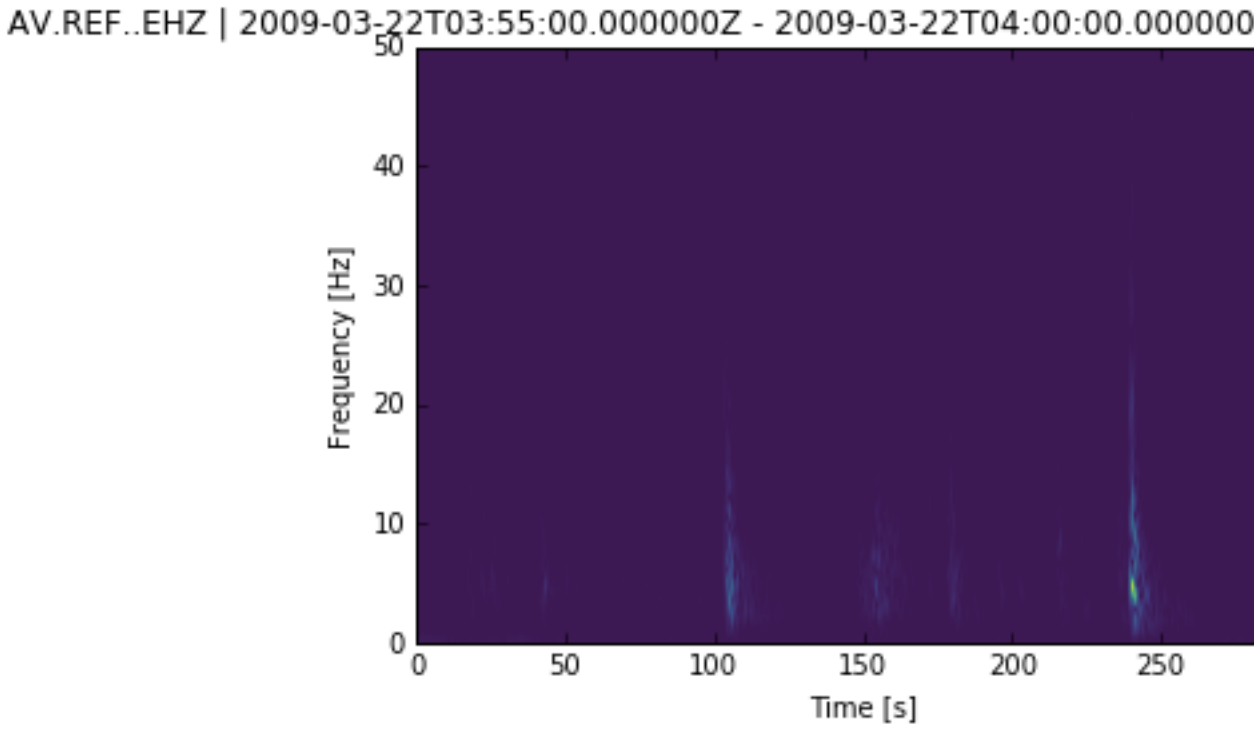
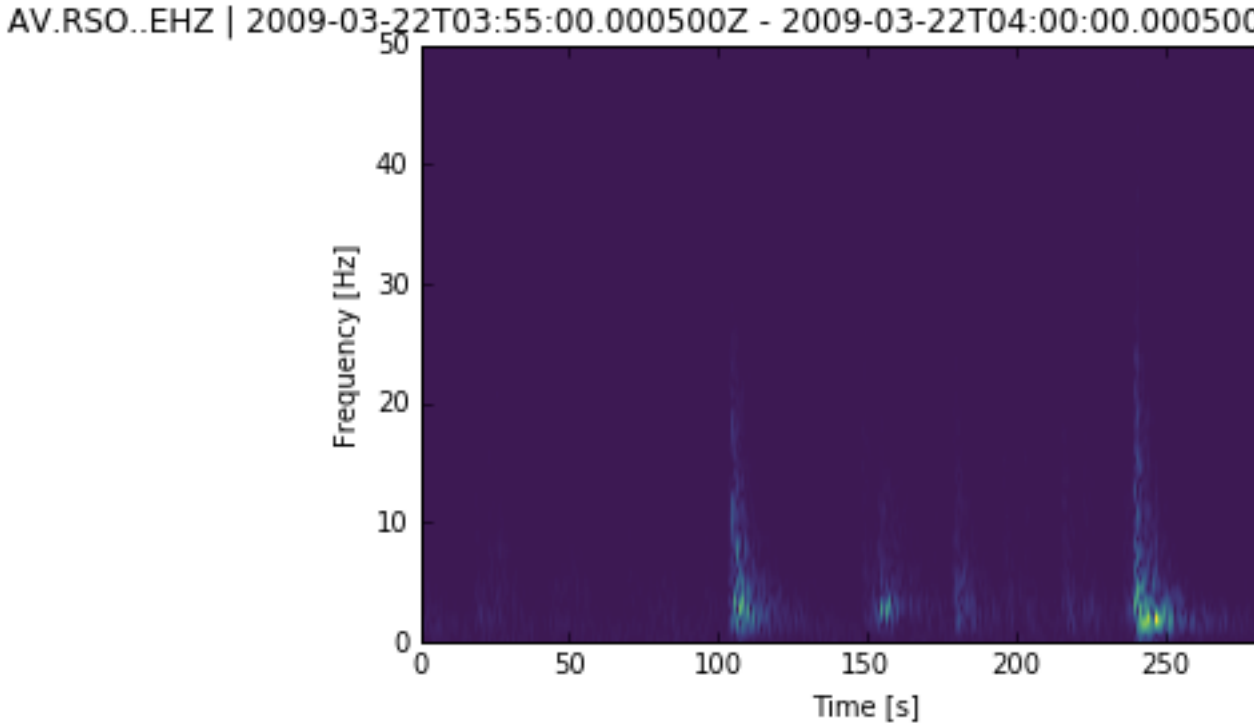
```
In [20]: import sys
    ...:
sys.path.append('/Users/glennthompson/Dropbox/scratch_matlab')
    ...: import tune_sta_lta as tsl
    ...: from obspy.core import read
    ...: import obspy.signal.trigger as trigger
    ...: from obspy.core.utcdatetime import UTCDateTime
    ...: tstart = UTCDateTime(2009, 3, 22, 3, 55, 0)
    ...: tend = UTCDateTime(2009, 3, 22, 4, 0, 0)
    ...: st =
read("/Users/glennthompson/Dropbox/scratch_matlab/SEEDDATA/R*2009
.081", starttime=tstart, endtime=tend)
    ...: st.plot(type='relative', equal_scale=False)
    ...: st.spectrogram()
```


Time in seconds relative to 2009-03-22T03:55:00
AV.REF..EHZ
8000
4000
0
−4000
−8000
AV.REF..EHN
6000
3000
0
−3000
−6000
AV.REF..EHE
9000
6000
3000
0
−3000
−6000
AV.RSO..EHZ
2000
1000
0
−1000


−2000
AV.REF..EHE | 2009-03-22T03:55:00.000000Z - 2009-03-22T04:00:00.000000
Frequency [Hz]
Time [s]
0
10
20
30
40
50
0
50
100
150
200
250

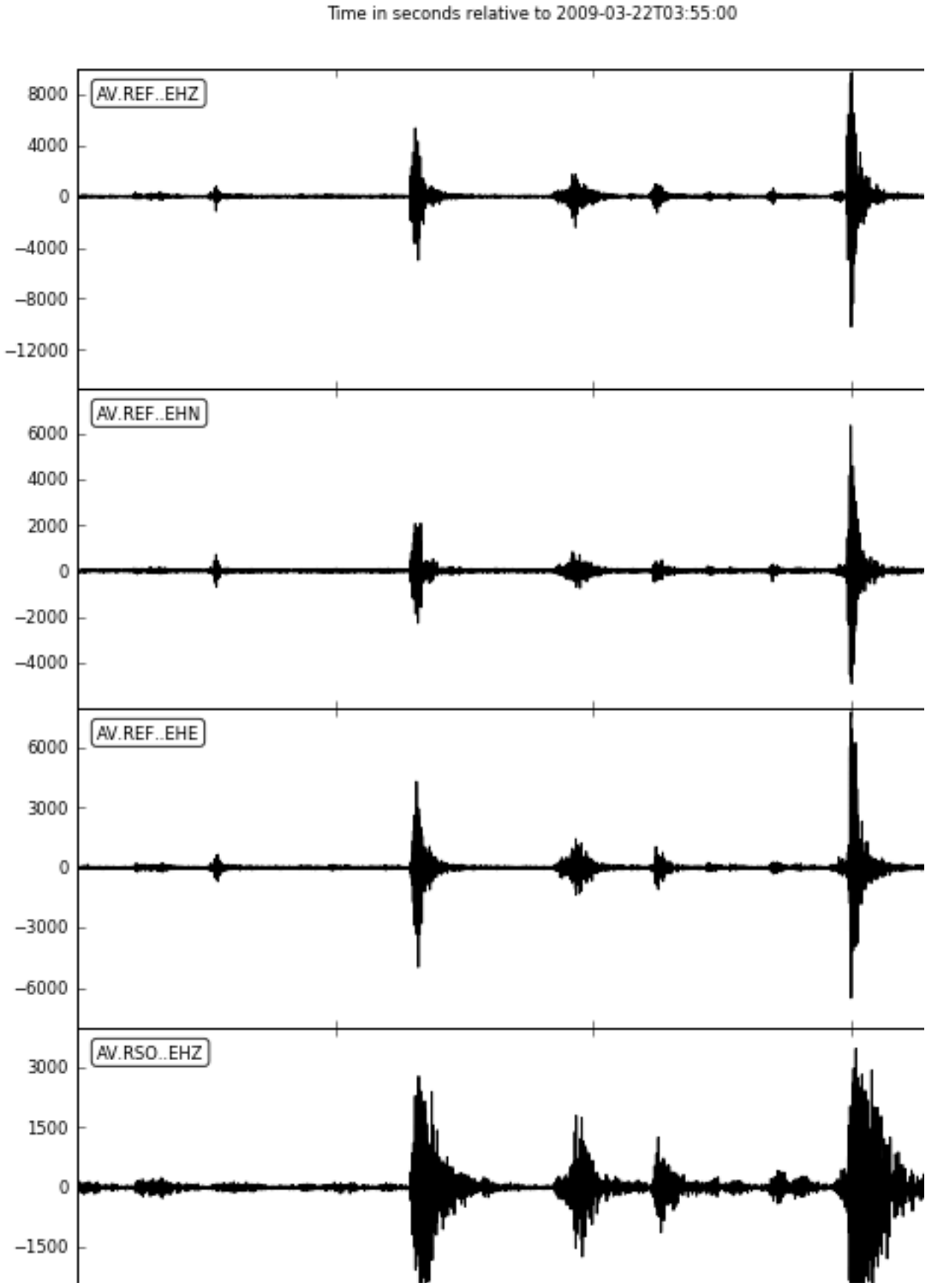

AV.REF..EHN | 2009-03-22T03:55:00.000000Z - 2009-03-22T04:00:00.000000
Frequency [Hz]
Time [s]
0
10
20
30
40
50
0
50
100
150
200
250

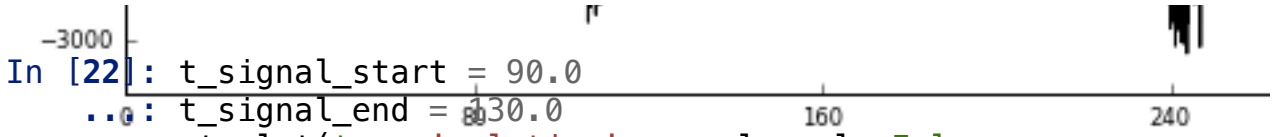




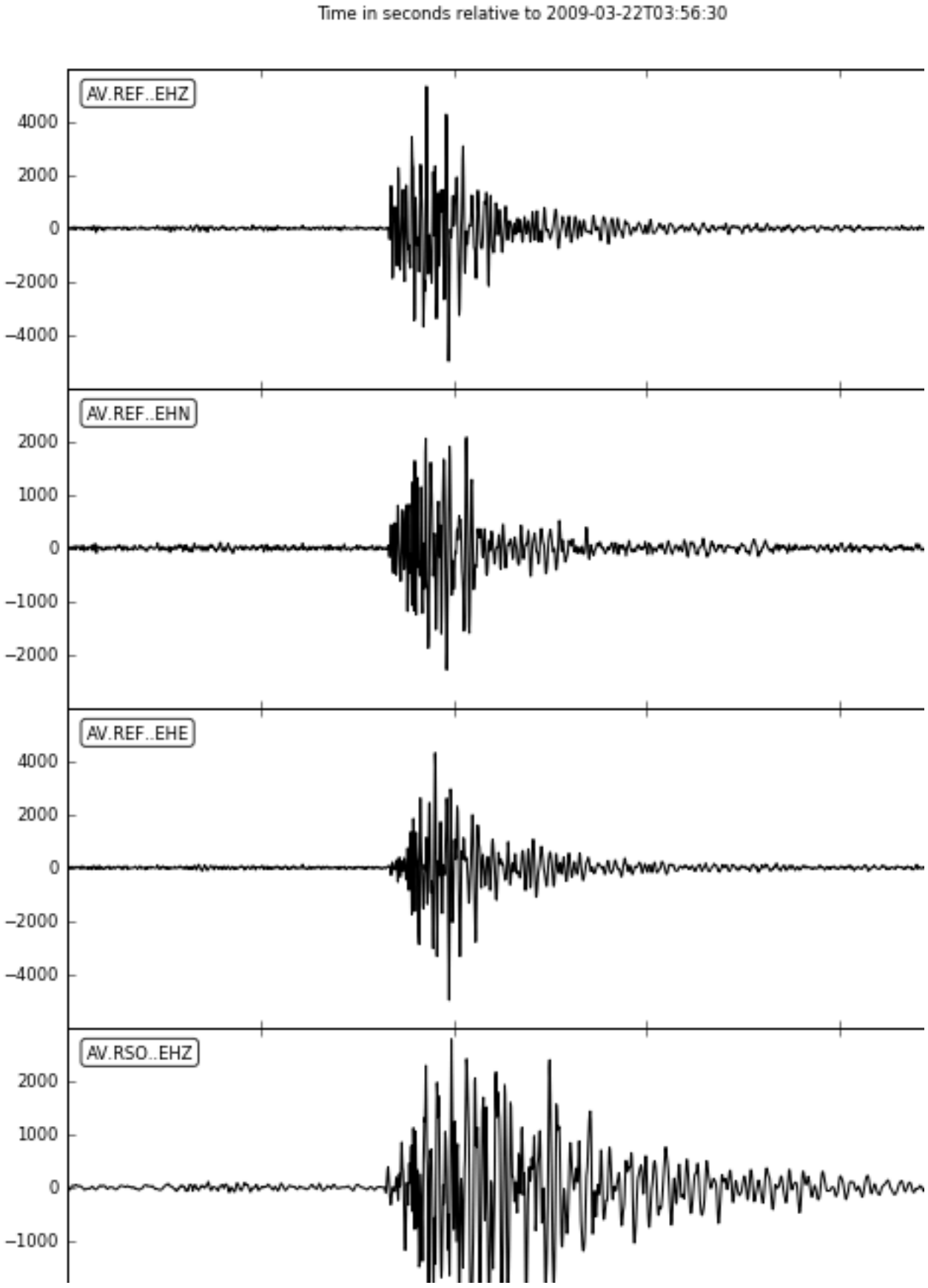

```
Out[20]: [None, None, None, None]

In [21]: st.filter('bandpass', freqmin=0.8, freqmax=12.0,
corners=2, zerophase=False)
    ...: st.decimate(factor=2, strict_length=False)
```

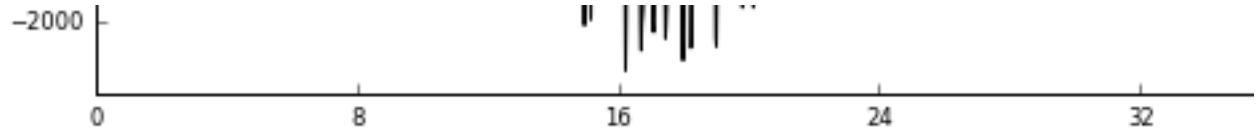
```
...: st.plot(type='relative', equal_scale=False)
```


Time in seconds relative to 2009-03-22T03:55:00
AV.REF..EHZ
8000
4000
0
−4000
−8000
−12000
AV.REF..EHN
6000
4000
2000
0
−2000
−4000
AV.REF..EHE
6000
3000
0
−3000
−6000
AV.RSO..EHZ
3000
1500
0
−1500



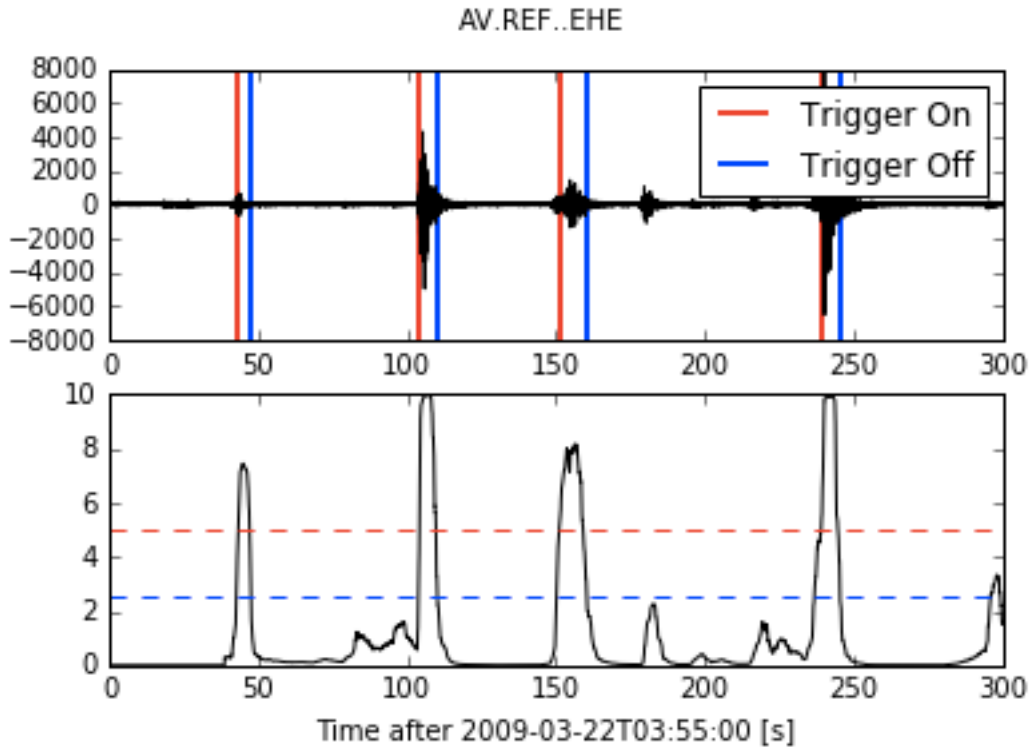

```
In [22]: t_signal_start = 90.0
    ...: t_signal_end = 130.0
    ...: st.plot(type='relative', equal_scale=False,
starttime=tstart+t_signal_start, endtime=tstart+t_signal_end)
    ...:
```

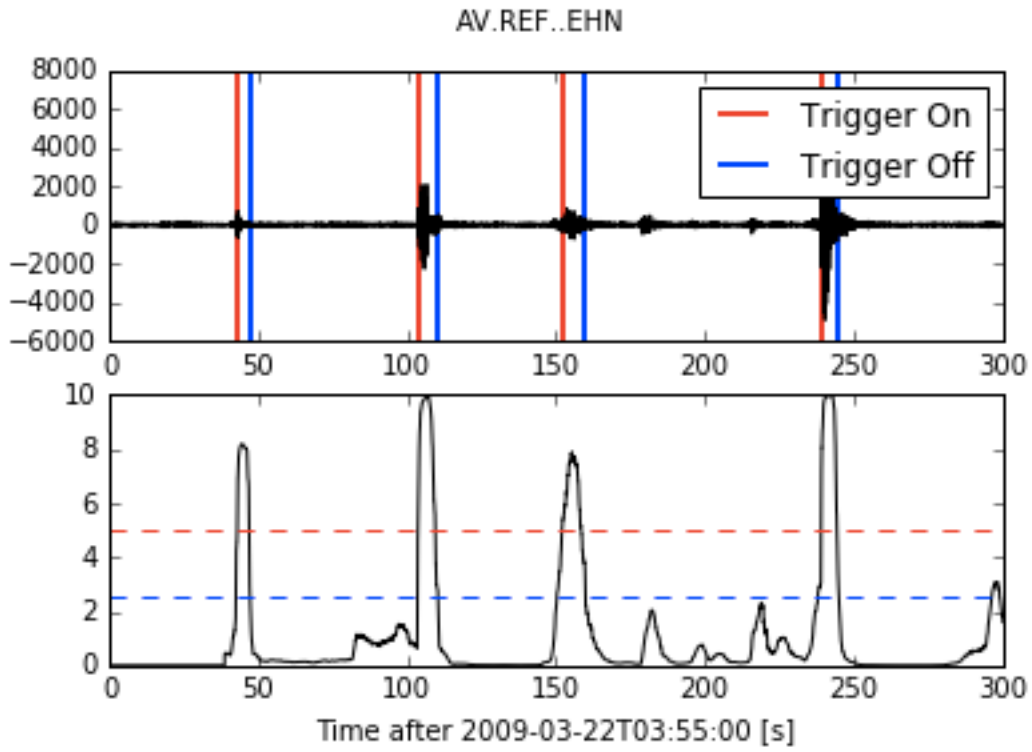

Time in seconds relative to 2009-03-22T03:56:30
AV.REF..EHZ
4000
2000
0
−2000
−4000
AV.REF..EHN
2000
1000
0
−1000
−2000
AV.REF..EHE
4000
2000
0
−2000
−4000
AV.RSO..EHZ
2000
1000
0
−1000

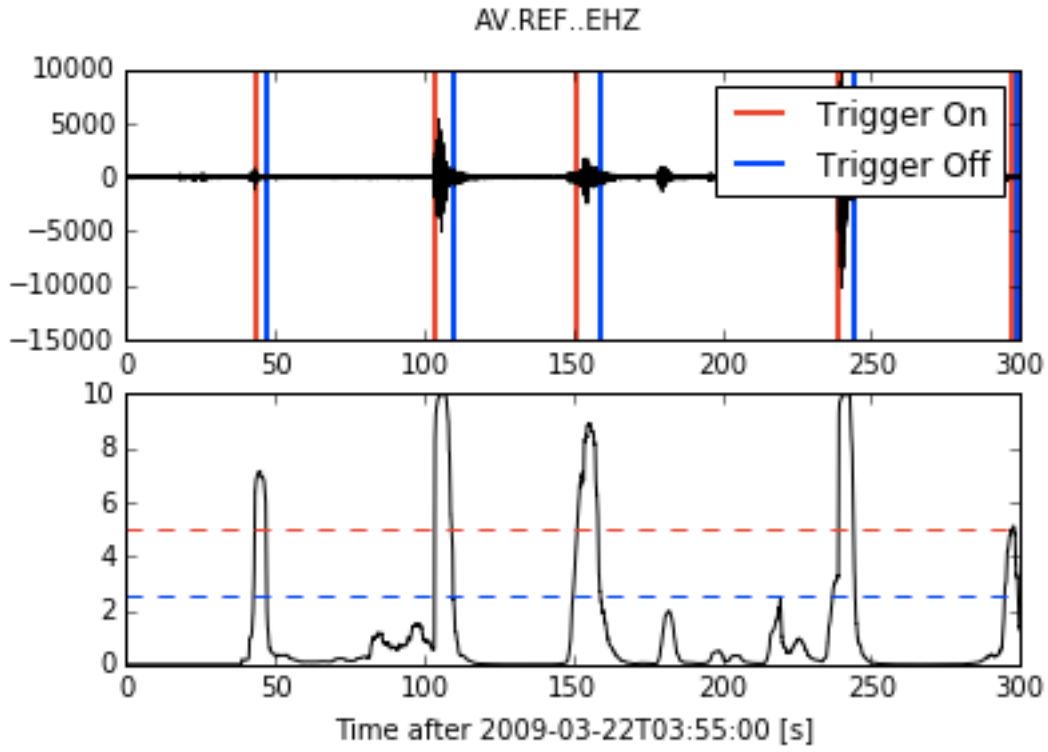


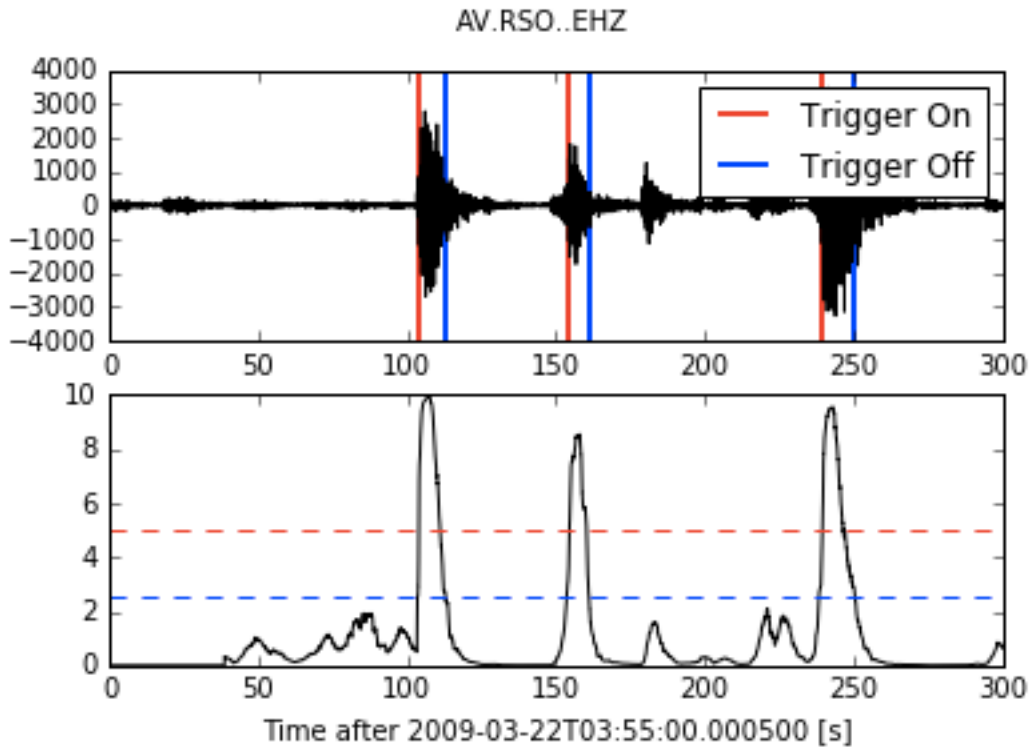

```
In [23]: algorithm = 'classic_sta_lta'
    ...: numtries = 30
    ...: sta_best = list()
    ...: lta_best = list()
    ...: for tr in st:
    ...:     result = tsl.tune_sta_lta(tr, algorithm,
t_signal_start, t_signal_end, numtries )
    ...:     sta_best.append(result[0])
    ...:     lta_best.append(result[1])
    ...:
    ...:

Algorithm: classic_sta_lta
sta_seconds=5.1 lta_seconds=15.3 max(staltaratio)=3.0
sta_seconds=8.0 lta_seconds=32.0 max(staltaratio)=4.0
sta_seconds=9.8 lta_seconds=68.6 max(staltaratio)=6.9
sta_seconds=3.4 lta_seconds=34.0 max(staltaratio)=9.9

Algorithm: classic_sta_lta
sta_seconds=7.6 lta_seconds=30.4 max(staltaratio)=4.0
sta_seconds=8.9 lta_seconds=53.4 max(staltaratio)=5.9
sta_seconds=3.1 lta_seconds=21.7 max(staltaratio)=7.0
sta_seconds=1.3 lta_seconds=11.7 max(staltaratio)=8.8
sta_seconds=6.1 lta_seconds=54.9 max(staltaratio)=8.9
sta_seconds=5.5 lta_seconds=55.0 max(staltaratio)=9.9

Algorithm: classic_sta_lta
sta_seconds=2.0 lta_seconds=14.0 max(staltaratio)=7.0
sta_seconds=6.0 lta_seconds=60.0 max(staltaratio)=9.9
sta_seconds=4.3 lta_seconds=43.0 max(staltaratio)=9.9
sta_seconds=2.5 lta_seconds=25.0 max(staltaratio)=10.0

Algorithm: classic_sta_lta
sta_seconds=2.6 lta_seconds=13.0 max(staltaratio)=5.0
sta_seconds=0.8 lta_seconds=8.0 max(staltaratio)=9.4
sta_seconds=4.8 lta_seconds=48.0 max(staltaratio)=9.9

In [24]: import scipy.stats.mstats as mstats
    ...: sta_gmean=mstats.gmean(sta_best)
    ...: lta_gmean=mstats.gmean(lta_best)
    ...: print sta_best
```

```
    ...: print lta_best
    ...: print "Best STA window = %.1f seconds, Best LTA window =
%.1f seconds" % (sta_gmean, lta_gmean)
    ...:
[3.4, 5.5, 2.5, 4.8]
[34.0, 55.0, 25.0, 48.0]
Best STA window = 3.9 seconds, Best LTA window = 38.7 seconds

In [25]: thresh_on = 5
    ...: thresh_off = 2.5
    ...: df = st[0].stats.sampling_rate
    ...: for tr in st:
    ...:     staltaratio = trigger.classic_sta_lta(tr.data,
int(sta_gmean * df), int(lta_gmean * df))
    ...:     trigger.plot_trigger(tr, staltaratio, thresh_on,
thresh_off)
    ...:
    ...:
```



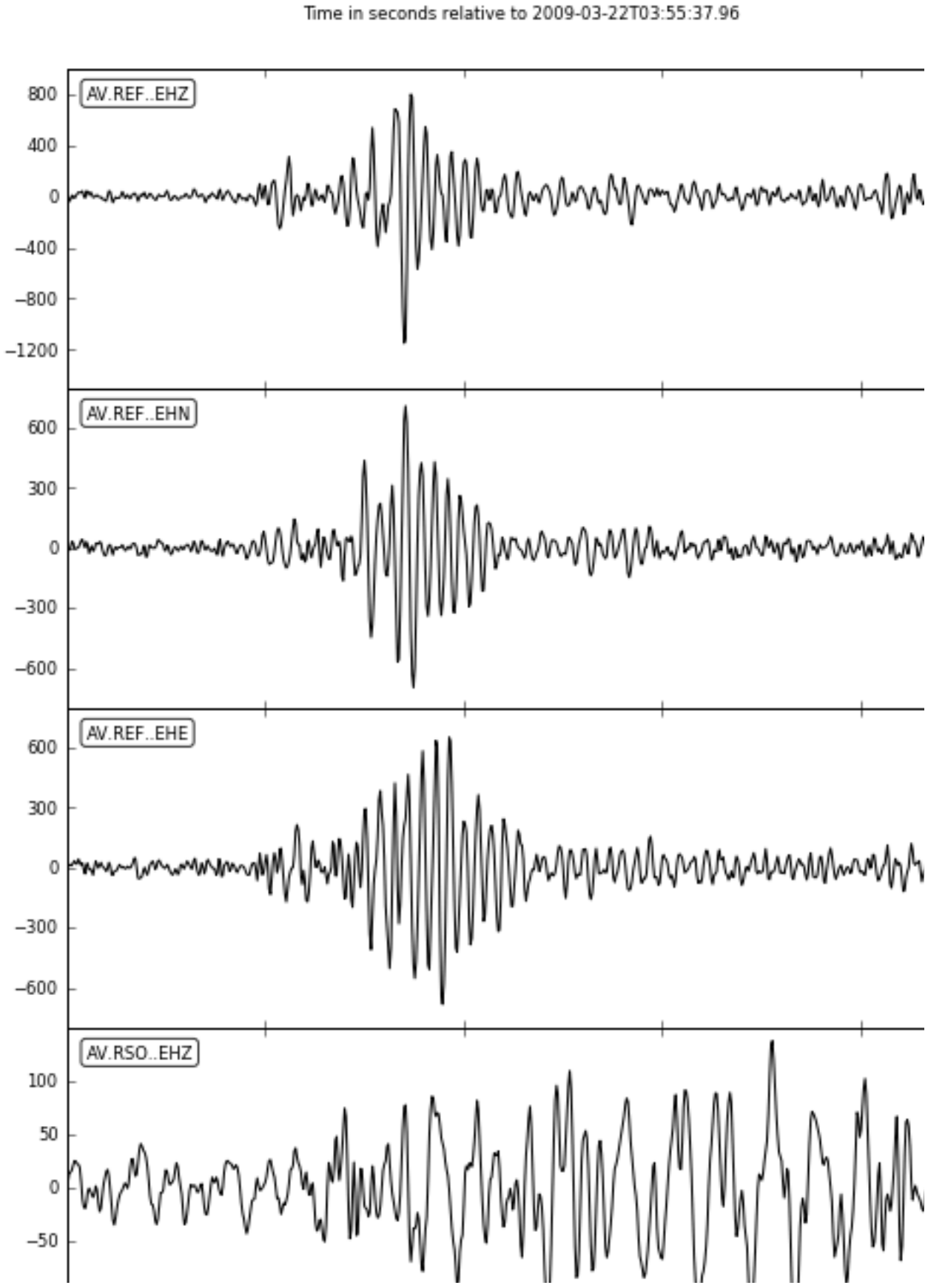

AV.REF..EHN
Trigger On
Trigger Off
Time after 2009-03-22T03:55:00 [s]


AV.REF..EHZ
Trigger On
Trigger Off
Time after 2009-03-22T03:55:00 [s]

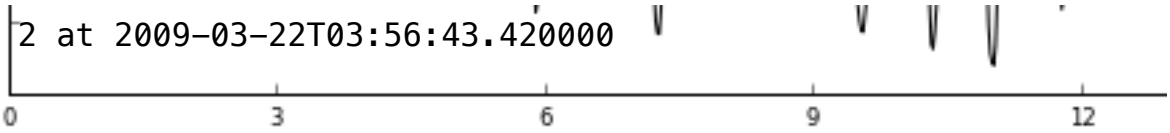


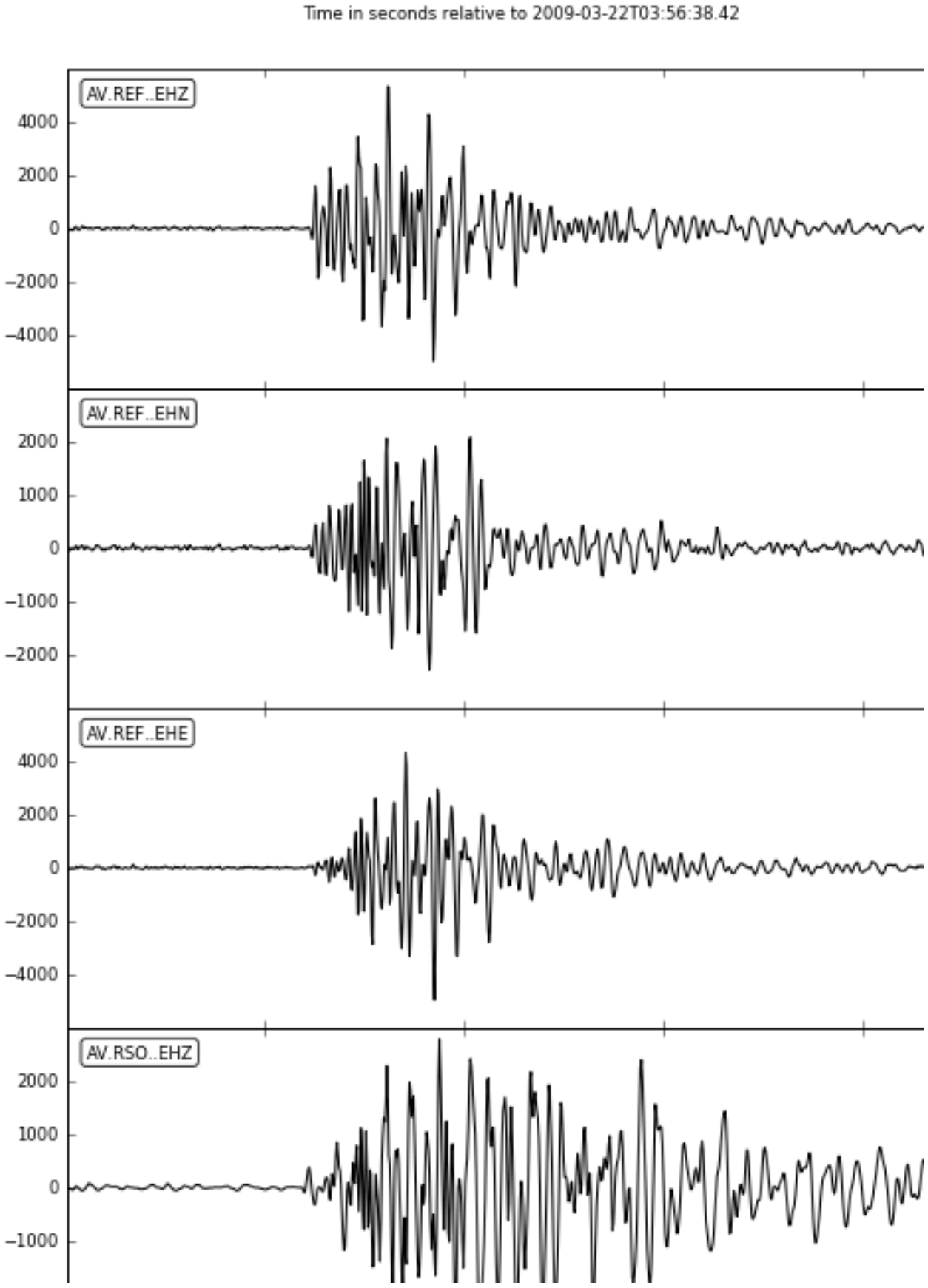

```
In [26]: triggers_per_event = 3
    ...: import re # for some dumb reason, coincidence trigger
needs algorithm name without the underlines!
    ...: algorithm_without_underlines = re.sub('_', '',
algorithm)
    ...: trig =
trigger.coincidence_trigger(algorithm_without_underlines,
thresh_on, thresh_off, st, triggers_per_event, sta=sta_gmean,
lta=lta_gmean)
    ...: from pprint import pprint
    ...: pprint(trig)
    ...: print "Number of events detected = %d" % len(trig)
[{u'coincidence_sum': 3.0,
  u'duration': 4.599999904632568,
  u'similarity': {},
  u'stations': [u'REF', u'REF', u'REF'],
  u'time': UTCDateTime(2009, 3, 22, 3, 55, 42, 960000),
  u'trace_ids': [u'AV.REF..EHN', u'AV.REF..EHZ',
u'AV.REF..EHE']},
 {u'coincidence_sum': 4.0,
  u'duration': 9.440500020980835,
  u'similarity': {},
  u'stations': [u'REF', u'REF', u'RSO', u'REF'],
  u'time': UTCDateTime(2009, 3, 22, 3, 56, 43, 420000),
```

```
  u'trace_ids': [u'AV.REF..EHZ',
                 u'AV.REF..EHN',
                 u'AV.RSO..EHZ',
                 u'AV.REF..EHE']},
 {u'coincidence_sum': 4.0,
  u'duration': 10.020500183105469,
  u'similarity': {},
  u'stations': [u'REF', u'REF', u'REF', u'RSO'],
  u'time': UTCDateTime(2009, 3, 22, 3, 57, 31, 20000),
  u'trace_ids': [u'AV.REF..EHE',
                 u'AV.REF..EHZ',
                 u'AV.REF..EHN',
                 u'AV.RSO..EHZ']},
 {u'coincidence_sum': 4.0,
  u'duration': 11.380500078201294,
  u'similarity': {},
  u'stations': [u'REF', u'REF', u'REF', u'RSO'],
  u'time': UTCDateTime(2009, 3, 22, 3, 58, 58, 980000),
  u'trace_ids': [u'AV.REF..EHZ',
                 u'AV.REF..EHE',
                 u'AV.REF..EHN',
                 u'AV.RSO..EHZ']}]
Number of events detected = 4

In [27]: pretrig = 5;
    ...: posttrig = 5;
    ...: count = 0
    ...: for thistrig in trig:
    ...:     count += 1
    ...:     print "Event %d at %s" %
(count,thistrig['time'].isoformat())
    ...:     st2 = st.copy()
    ...:     st2.trim(starttime = thistrig['time'] - pretrig,
endtime = thistrig['time'] + thistrig['duration'] + posttrig)
    ...:     st2.plot(type='relative', equal_scale=False)
    ...:
    ...:
Event 1 at 2009-03-22T03:55:42.960000
```

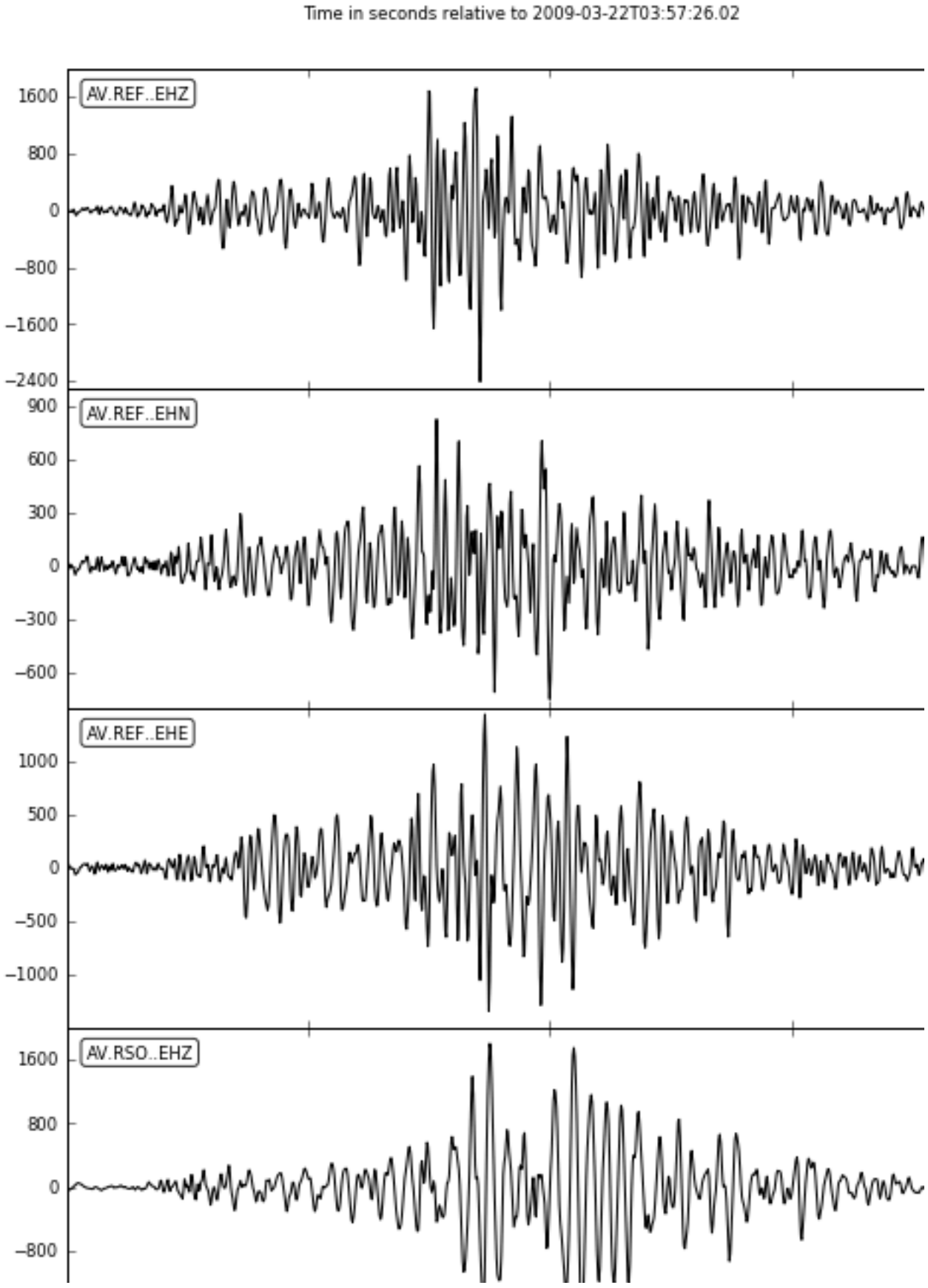

Time in seconds relative to 2009-03-22T03:55:37.96
AV.REF..EHZ
800
400
0
−400
−800
−1200
AV.REF..EHN
600
300
0
−300
−600
AV.REF..EHE
600
300
0
−300
−600
AV.RSO..EHZ
100
50
0
−50

Event 2 at 2009-03-22T03:56:43.420000

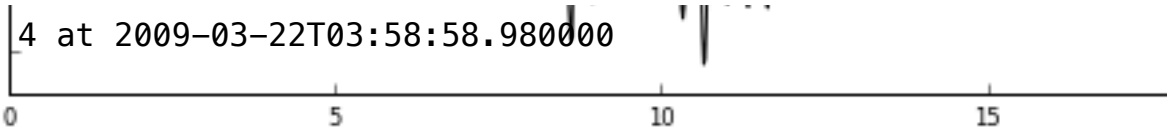


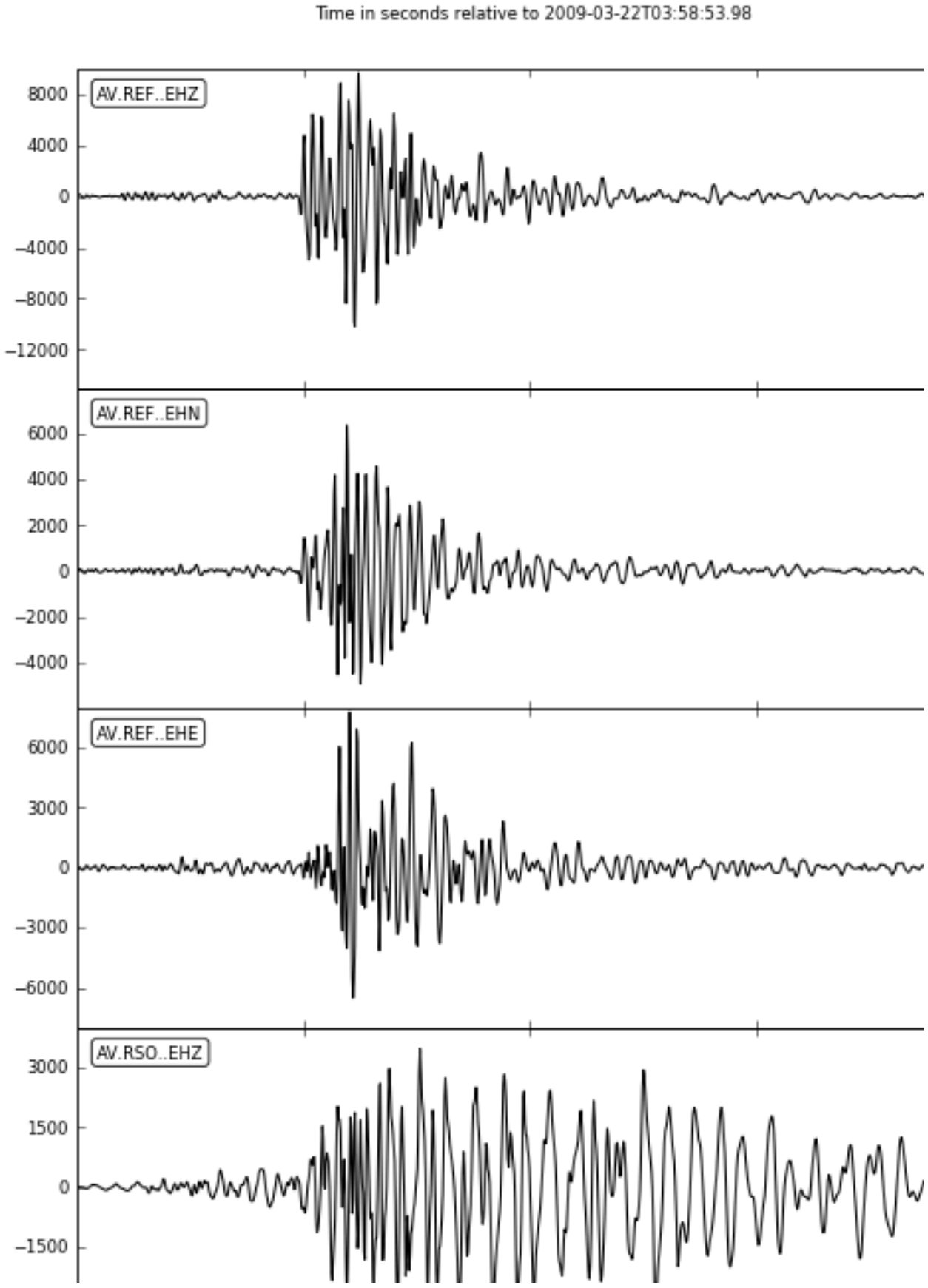

Time in seconds relative to 2009-03-22T03:56:38.42
AV.REF..EHZ
4000
2000
0
−2000
−4000
AV.REF..EHN
2000
1000
0
−1000
−2000
AV.REF..EHE
4000
2000
0
−2000
−4000
AV.RSO..EHZ
2000
1000
0
−1000

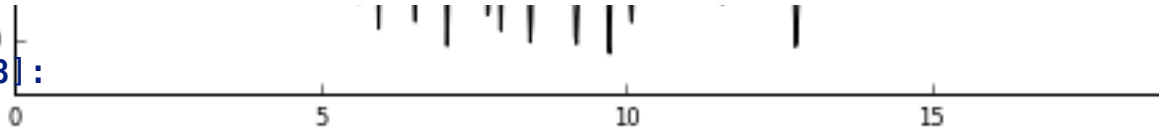
Event 3 at 2009-03-22T03:57:31.020000


Time in seconds relative to 2009-03-22T03:57:26.02
AV.REF..EHZ
1600
800
0
−800
−1600
−2400
AV.REF..EHN
900
600
300
0
−300
−600
AV.REF..EHE
1000
500
0
−500
−1000
AV.RSO..EHZ
1600
800
0
−800

Event 4 at 2009-03-22T03:58:58.980000




Time in seconds relative to 2009-03-22T03:58:53.98
AV.REF..EHZ
8000
4000
0
−4000
−8000
−12000
AV.REF..EHN
6000
4000
2000
0
−2000
−4000
AV.REF..EHE
6000
3000
0
−3000
−6000
AV.RSO..EHZ
3000
1500
0
−1500



In [28]: